

Visual Programming Languages

Lab Manual

[Fall 2019]

|  |  |
| --- | --- |
| Student Name: Adeel Nisar |  |
| Student Id: 13399 |  |
|  |  |

|  |  |
| --- | --- |
| Prepared By: *Adeel Nisar* |  |
| Instructor: *Dr. Noman Islam* |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **S. No** | **Date** | **Experiment** |  |
| **1** | \_\_/\_\_/\_\_ | To setup the environment and familiarize with C# |  |
| **2** | \_\_/\_\_/\_\_ | To study and understand how to write programs in C# using loops and arrays |  |
| **3** | \_\_/\_\_/\_\_ | To study and implement object oriented programming concepts in C# |  |
| **4** | \_\_/\_\_/\_\_ | To study and implement Windows Forms application in C# |  |
| **5** | \_\_/\_\_/\_\_ | To study and implement Collections in C# |  |
| **6** | \_\_/\_\_/\_\_ | To study and implement I/O in C# |  |
| **7** | \_\_/\_\_/\_\_ | To study and implement XML parsing in C# |  |
| **8** | \_\_/\_\_/\_\_ | To study and implement WPF and its layouts in C# |  |
| **9** | \_\_/\_\_/\_\_ | To study and implement LINQ in C# |  |
| **10** | \_\_/\_\_/\_\_ | To study and implement ADO .Net in C# |  |
|  |  |  |  |

**Lab 1: To setup the environment and familiarize with C#**

The objective of this lab is to set up the Visual Studio environment and get some familiarity with the C# language.

Download and install Visual Studio .Net. Visual Studio is the leading platform powered by Microsoft for development on .net framework.

**Lab Tasks:**

Write a small program in C# to print your CV.

Write a program to calculate whether an input number is even or odd.

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int a;

Console.WriteLine("enter any number:");

a = int.Parse(Console.ReadLine());

if (a % 2 == 0)

Console.WriteLine("number is even ");

else

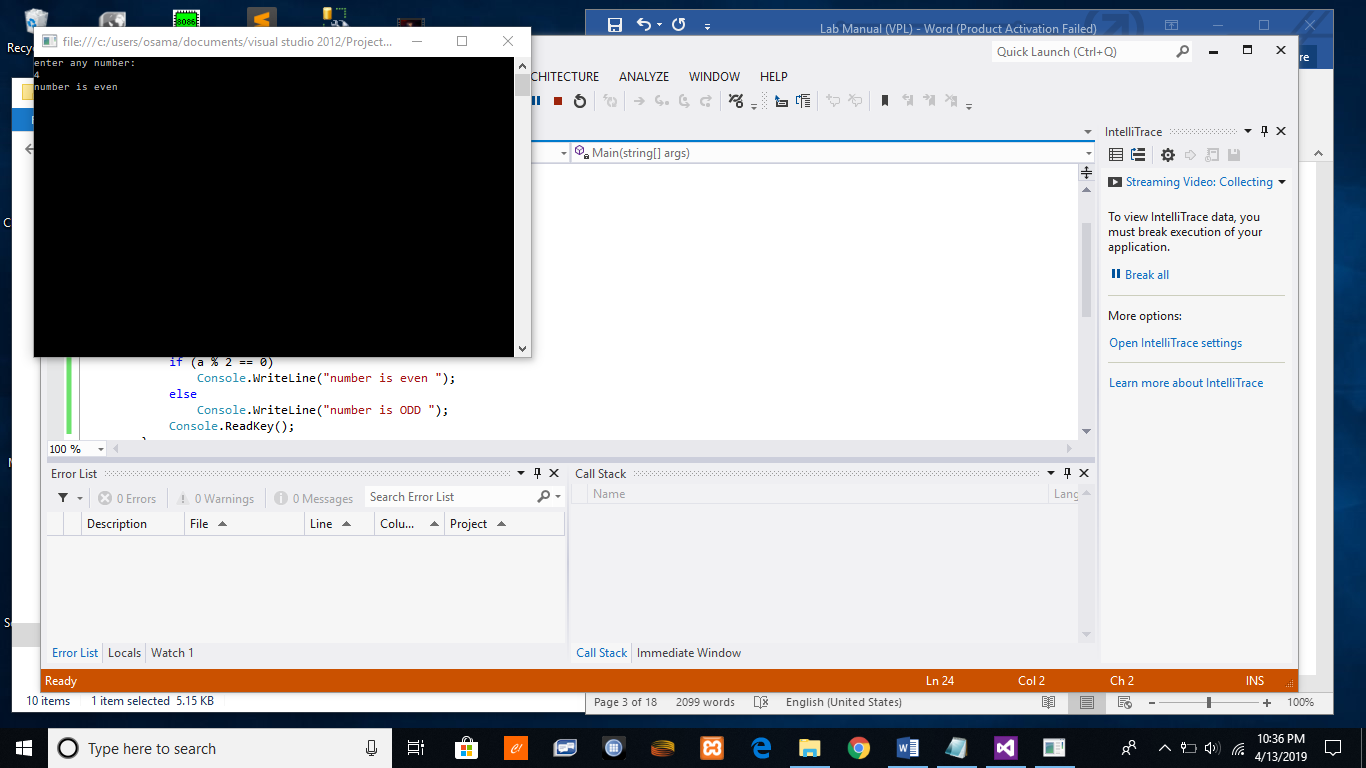
Console.WriteLine("number is ODD ");

Console.ReadKey(); }

}

}

Output:



Write a program that takes thee numbers from user as input. The program then prints the maximum and minimum of the input numbers.

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int no1;

int no2;

int no3;

Console.WriteLine("enter your first number:");

no1 = int.Parse(Console.ReadLine());

Console.WriteLine("enter your second number:");

no2 = int.Parse(Console.ReadLine());

Console.WriteLine("enter your second number:");

no3 = int.Parse(Console.ReadLine());

if (no1 > no2 && no1 > no3)

{

Console.WriteLine("no1 is greater" + no1);

}

else if (no2 > no3 && no2 > no1)

{

Console.WriteLine("no2 is greater" + no2);

}

else if (no3 > no1 && no3 > no2)

{

Console.WriteLine("no3 is greater" + no3); }

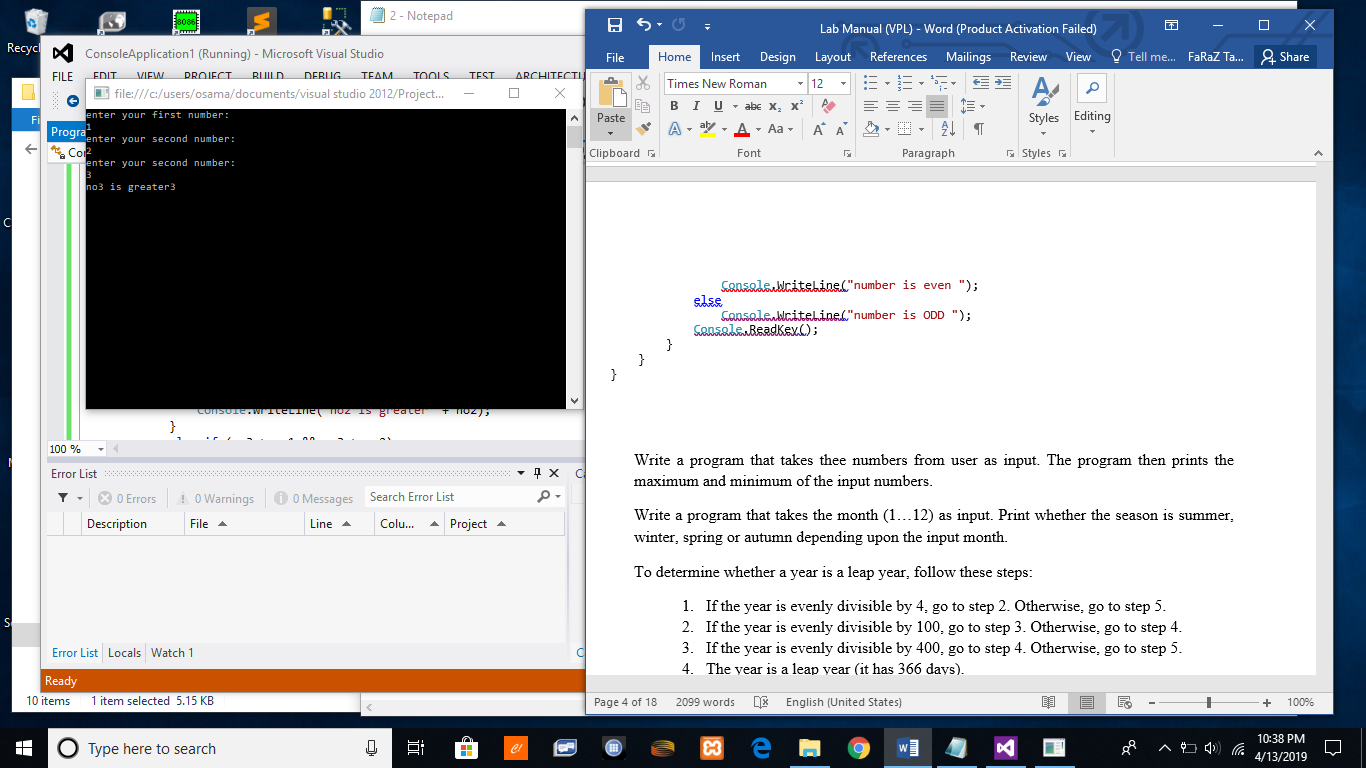
Console.ReadKey();

}

}

}

Output



Write a program that takes the month (1…12) as input. Print whether the season is summer, winter, spring or autumn depending upon the input month.

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int month;

Console.WriteLine("enter the month from 1-12");

month = int.Parse(Console.ReadLine());

if (month == 1 || month == 2 || month == 3 || month == 4)

{

Console.WriteLine("the season is winter");

}

if (month == 4 || month == 5)

{

Console.WriteLine("the season is summer");

}

if (month == 6 || month == 7 || month == 8)

{

Console.WriteLine("the season is spring");

}

if (month == 9 || month == 10 || month == 11 || month == 12)

{

Console.WriteLine("the season is autumn");

}

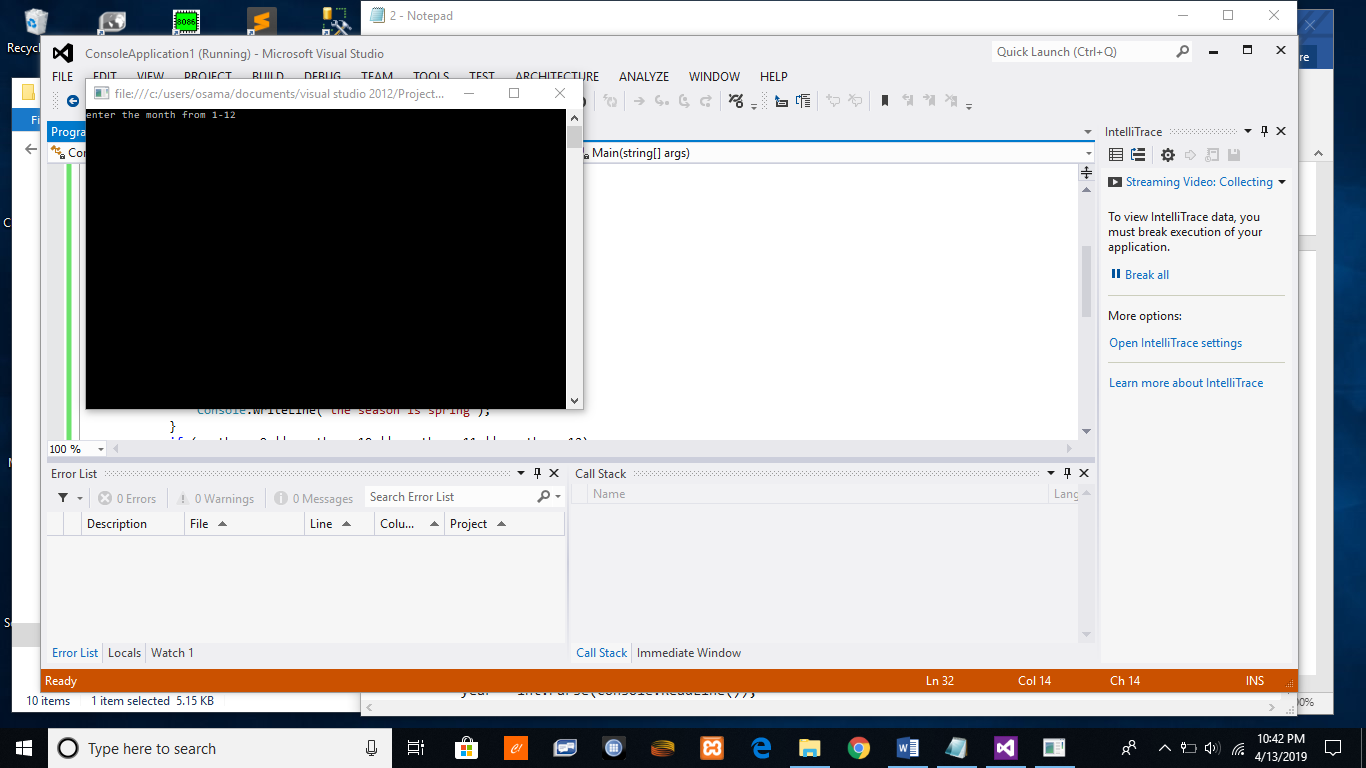
Console.ReadKey();

}

}

}

Output



To determine whether a year is a leap year, follow these steps:

* 1. If the year is evenly divisible by 4, go to step 2. Otherwise, go to step 5.
  2. If the year is evenly divisible by 100, go to step 3. Otherwise, go to step 4.
  3. If the year is evenly divisible by 400, go to step 4. Otherwise, go to step 5.
  4. The year is a leap year (it has 366 days).
  5. The year is not a leap year (it has 365 days).

Write a program to input an year as integer. Using if…else, determines whether the input is a leap year or not.

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

{

int year;

Console.WriteLine("enter the year");

year = int.Parse(Console.ReadLine());

if (year % 4 == 0)

{

if (year % 100 == 0)

{

Console.WriteLine("it is a leap year");

}

else

Console.WriteLine("it is not a leap year");

}

if (year % 4 == 0)

{

if (year % 100 == 0)

{

if (year % 400 == 0)

{

Console.WriteLine("it is a leapyear");

}

}

}

else

Console.WriteLine("its not a leap year");

}

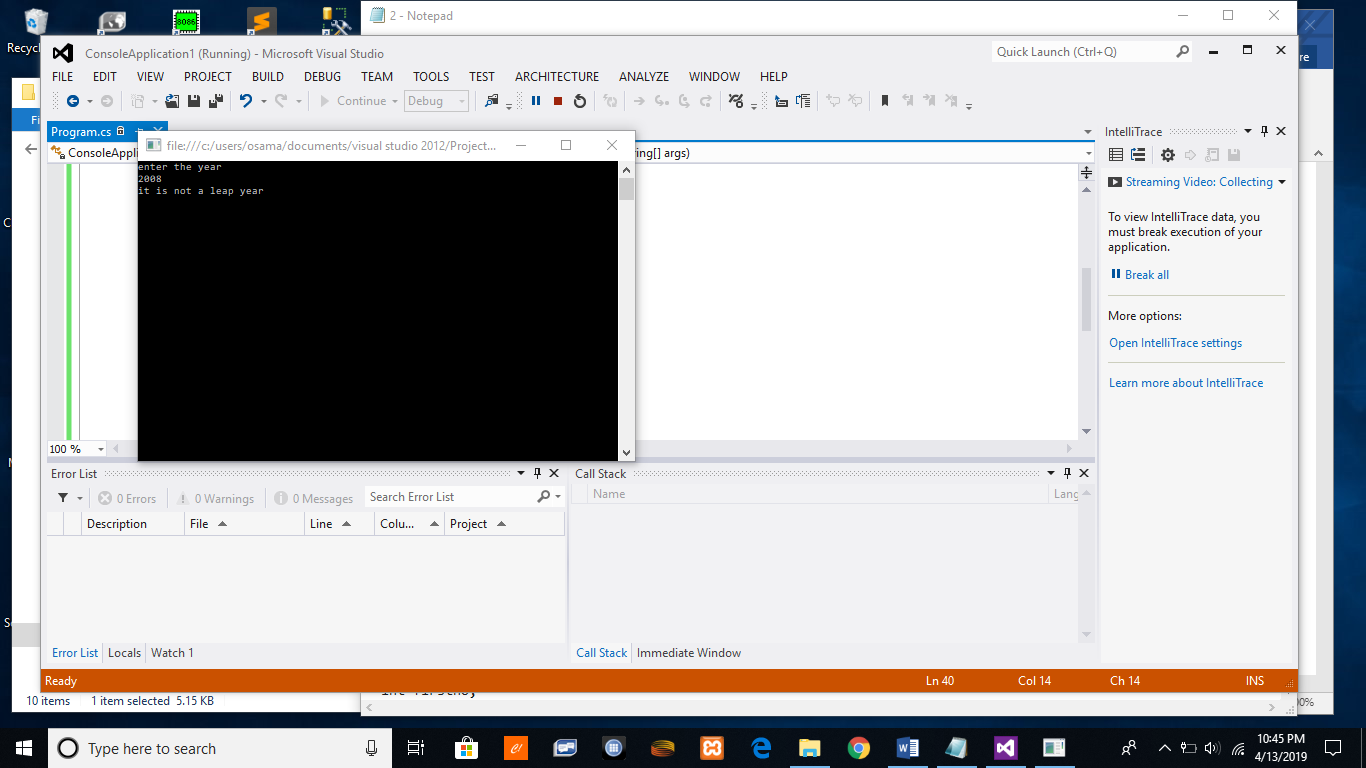
Console.ReadKey();

}

}

}

Output



Write a program that takes two numbers as input and an operator as input. Using the switch statement, the program should calculate the result when the operator is applied on the two input numbers.

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int firstno;

int secodno;

int total;

string op;

Console.WriteLine("enter your first number");

firstno = int.Parse(Console.ReadLine());

Console.WriteLine("enter your second number");

secodno = int.Parse(Console.ReadLine());

Console.WriteLine("enter your operator");

op = Console.ReadLine();

switch (op)

{

case "+": total = firstno + secodno;

Console.WriteLine("result=" + total);

break;

case "-": total = firstno - secodno;

Console.WriteLine("result=" + total);

break;

case "\*": total = firstno \* secodno;

Console.WriteLine("result=" + total);

break;

case "/": total = firstno / secodno;

Console.WriteLine("result=" + total);

break;

}

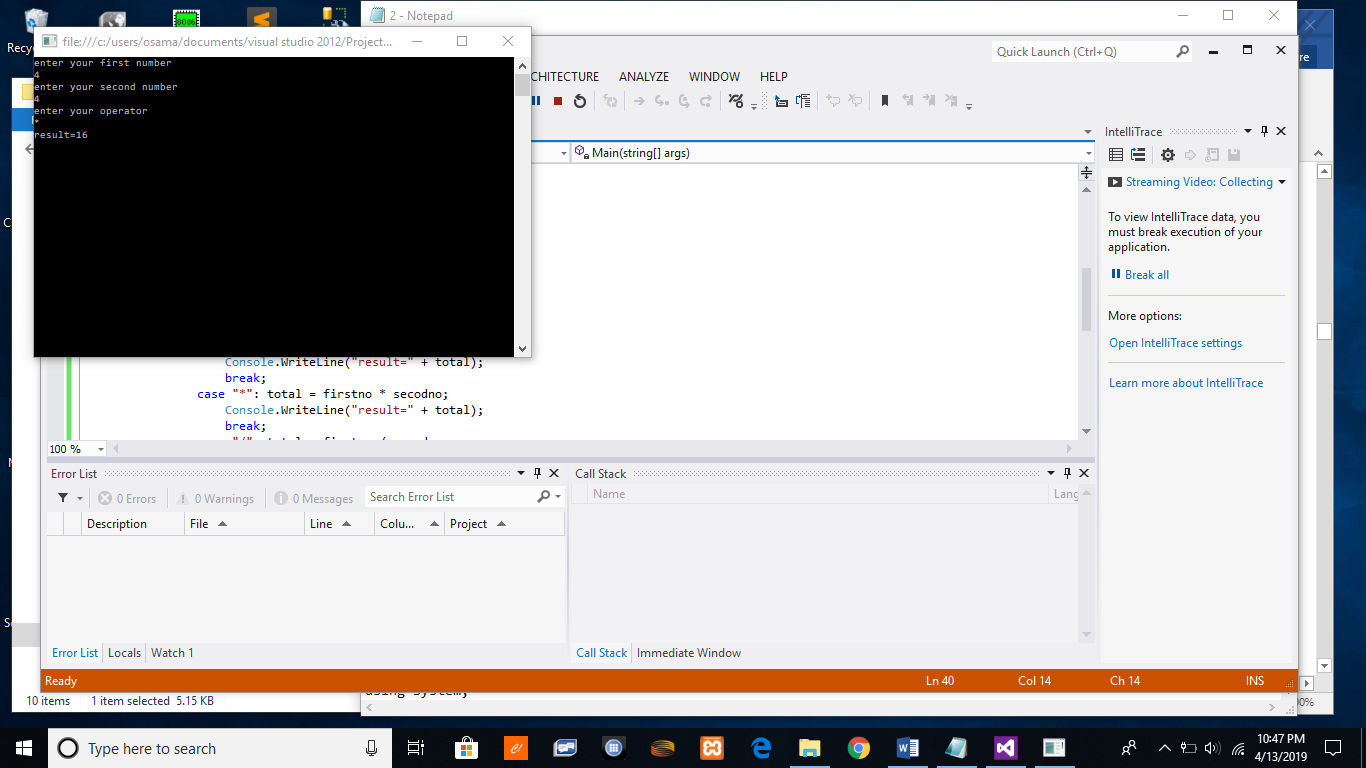
Console.ReadKey();

}

}

}

Output



Write a program to print Iqra University marks sheet using if…else statement

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

{

float firstno;

float secodno;

float thirdno;

float fourthno;

float fifthno;

float total;

float PER;

Console.WriteLine("enter your marks in computer programming");

firstno = int.Parse(Console.ReadLine());

Console.WriteLine("enter your marks in VPL");

secodno = int.Parse(Console.ReadLine());

Console.WriteLine("enter your marks in WPL");

thirdno = int.Parse(Console.ReadLine());

Console.WriteLine("enter your marks in MP");

fourthno = int.Parse(Console.ReadLine());

Console.WriteLine("enter your marks in OS");

fifthno = int.Parse(Console.ReadLine());

Console.WriteLine("Marksheet");

Console.WriteLine("CP:"+firstno);

Console.WriteLine("VPL:"+secodno);

Console.WriteLine("WPL:"+thirdno);

Console.WriteLine("MP:"+fourthno);

Console.WriteLine("OS:"+fifthno);

total = firstno + secodno + thirdno + fourthno + fifthno;

Console.WriteLine("TOTAL MARKS "+ total +" OUT OF 500");

PER = (total / 500) \* 100;

Console.WriteLine("PERCENTAGE="+PER);

}

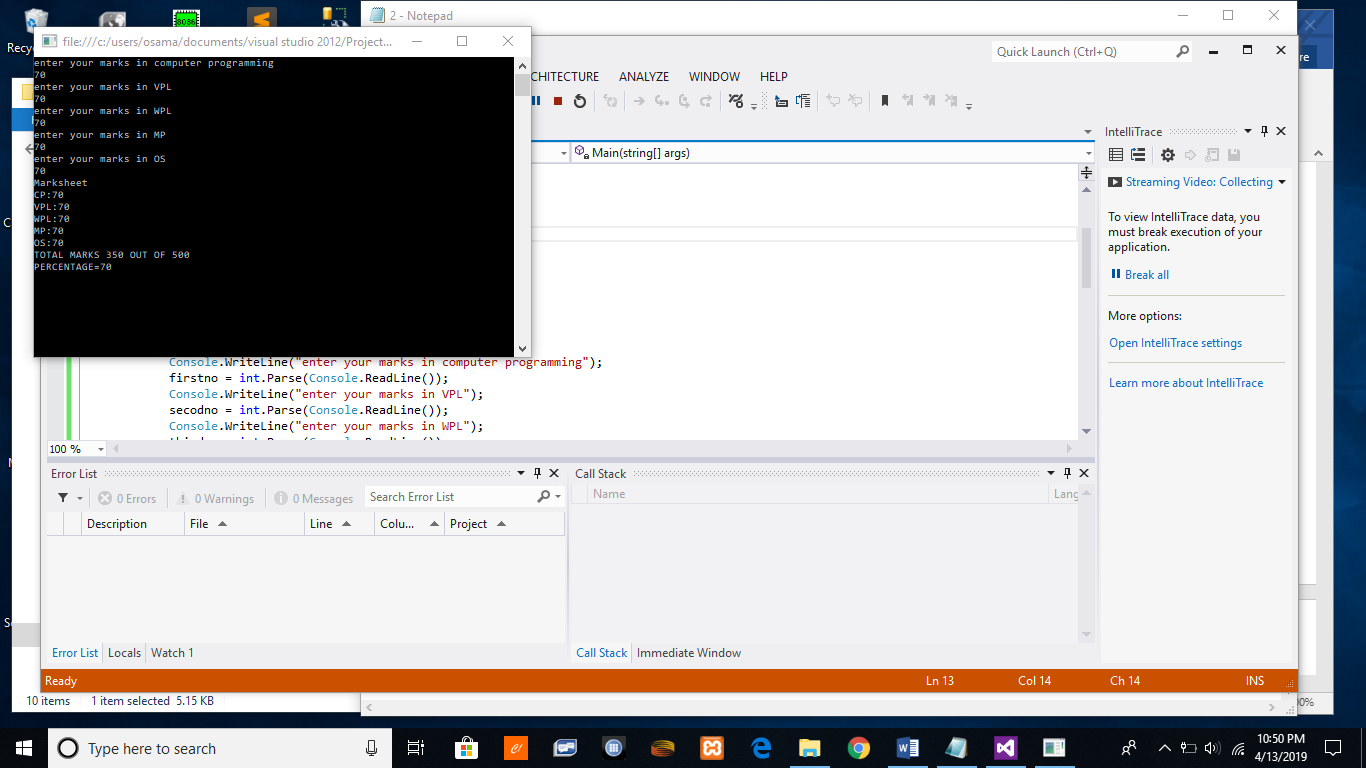
Console.ReadKey();

}

}

}

Output



**Lab 2: To study and understand how to write programs in C# using loops, arrays and other constructs**

The objective of this lab is to start writing programs in C# using its basic constructs such as loops, conditions, arrays etc. Following are intrinsic data types supported by C#.

**Lab Tasks:**

1. Write a program to count the frequency of each element of an array.
2. Write a program to find maximum and minimum element in an array.
3. Write a program to separate odd and even integers in separate array
4. Write a program to find the length of a string without using library function.
5. Write a program to count the total number of words in a string.
6. Write a program to create a recursive function to calculate the Fibonacci number of a specific term.

**Code:**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication6

{

class Fibonacci

{

static ArrayList al = new ArrayList();

static void Main(string[] args)

{

f(10);

}

static int f(int n)

{

if(n<=1)

{

return n;

}

int result = f(n - 2) + f(n - 1);

if (al.IndexOf(result) == -1)

{

Console.Write(" "+ result + " ");

al.Add(result);

}

return result;

}

}

}

**Lab 3: To study and implement object oriented programming concepts in C#**

Object-oriented programming (OOP) refers to a type of computer programming (software design) in which programmers define not only the data type of a data structure, but also the types of operations (functions) that can be applied to the data structure. In this lab, we are going to implement the three basic pillars of OOP i.e. encapsulation, inheritance and polymorphism.

**Tasks:**

1. Encapsulation
   * + 1. Create a class Circle.
       2. Define a property named radius. In the set method, check if the radius is negative then throw an exception System.ArgumentException.
       3. Define a one argument constructor to initialize the radius.
       4. Define a no argument constructor to initialize the radius value to zero using constructor chaining.
       5. Define a method GetArea() to calculate the area of circle.
       6. Create a Test class.
       7. In the Main method, define two objects of Circle and initialize them with random values.
       8. Call the GetArea() method of each object and print the area.
2. Inheritance/ Polymorphism
   * + 1. Create an abstract class Animal. Define a property: name of type string. Define an abstract method sound(). Define a constructor to initialize the animal’s name.
       2. Now define two abstract classes named Mammal and Non-Mammal that inherits the Animal class.
       3. Inherit the Mammal class to define Cat class. Implement the method sound that prints ‘Meow’.
       4. Implement classes for the Goat inheriting the Mammal and the Fish inheriting the Non-Mammal class
       5. Create a Test class. Define few objects of classes Cat, Goat and Fish. Assign the instance variables to reference variable of Animal class and polymorphically call them.

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication7

{

class Program

{

static void Main(string[] args)

{

Circle c = new Circle();

//c.radius = 10;

Console.WriteLine(c.radius);

}

}

}

**Code circle class:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication7

{

class Circle

{

private float r;

public float radius

{

get

{

return r;

}

set

{

this.r = value;

}

}

}

}

**Lab 4: To study and implement Windows Forms application in C#**

In this lab we will implement windows forms in C#. Windows Forms (WinForms) is a graphical (GUI) class library included as a part of Microsoft .NET Framework or Mono Framework, providing a platform to write rich client applications for desktop, laptop, and tablet PCs.

We will look at some of the most common controls used in C# and use them to develop a basic application.

**Tasks:**

1. Design a basic Sign Up form with fields name, id, password, email, address, gender and date of birth.
2. Add a checkbox to accept license agreement, and a view button to view the license agreement.
3. Add a Sign Up button.
4. When the user clicks on Sign Up, following validations are to be performed:
   1. Password must contain a capital letter and one digit
   2. Email address should be in proper format
   3. All the fields must be filled by user
   4. The user must be 18 years or above
   5. The user has accepted the license agreement
5. When the user clicks on view license agreement button, a new form is to be displayed with the license agreement.

Code:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace signup

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

int id = int.Parse(txtid.Text);

string name = txtname.Text;

string email = txtemail.Text;

string password = txtpass.Text;

string country = cmbcountry.Text;

string date = dateTimePicker1.Text;

string gender;

if (rdomail.Checked)

{

gender = "Male";

}

else { gender = "Female"; }

MessageBox.Show(gender);

if (chk1.Checked)

{

MessageBox.Show("plz fill the values");

}

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void btnview\_Click(object sender, EventArgs e)

{

Form2 obj = new Form2();

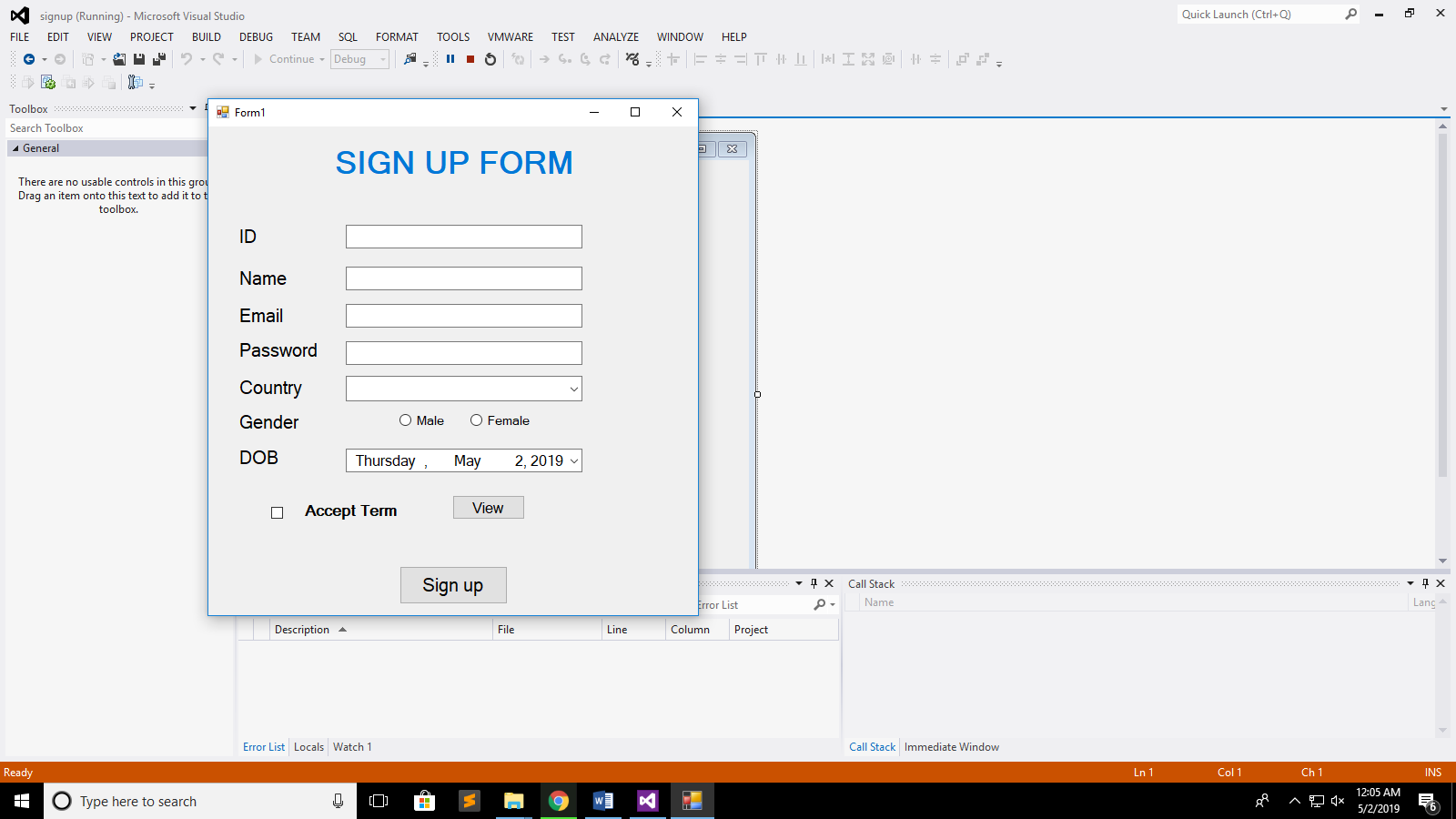
obj.Show();

}

}

}

**Output:**



**Lab 5: To study and implement Collections in C#**

In todays, lab we will implement the collections in C#. A collection is an abstract data type for grouping together multiple values. It's therefore sometime known as container. A collection is just a grouping of some objects with the same type.

**Lab Tasks:**

1. Create a WinForm in C# with following buttons: Add a Student, Edit a Student, Delete a Student, Search a Student.
2. Now, create a class Student with the following properties: name, age, gender, cgpa
3. Implement the Add a Student button. When the user clicks this button, open a new form from which user can provide the details. Save the results in a collection.
4. Implement the Edit button such that user can modify the students record. The changes are reflected in the collection.
5. Implement the Delete and Search options.

**Lab 6: To study and implement I/O in C#**

In this lab, we will use StreamReader and StreamWriter of C# to develop a basic application that can insert student’s records in a file. A stream can be defined as a sequence of data. The StreamReader is used to read data from a source and the StreamWriter is used for writing data to a destination.

**Lab Tasks:**

1. Create a basic form in C# with the following fields: student id, age, gender, marks.
2. Add the buttons to add a new record, delete a record and update a record, in memory.
3. Create a Menu Strip from which user can save the records in a file or load the records from a file.

Code:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

namespace lab6

{

public partial class Form1 : Form

{

List<Student> students = new List<Student>();

int current =-1;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

Student std = new Student();

std.id = textBox1.Text;

std.age = textBox2.Text;

if (Male.Checked)

{

std.gender = "male";

}

else {

std.gender = "female";

}

std.marks = textBox3.Text;

students.Add(std);

current = students.Count - 1;

MessageBox.Show("Student added successfully");

}

private void button5\_Click(object sender, EventArgs e)

{

if (current >= students.Count-1)

{

return;

}

else {

current++;

Student s = students[current];

textBox1.Text = s.id;

textBox2.Text = s.age;

textBox3.Text = s.marks;

if (s.gender == "male")

{

Male.Checked = true;

}

else

{

Female.Checked = true; } }

}

private void button4\_Click(object sender, EventArgs e)

{

if (current <= 0)

{

return;

}

else {

current--;

Student s = students[current];

textBox1.Text = s.id;

textBox2.Text = s.age;

textBox3.Text = s.marks;

if (s.gender == "male")

{

Male.Checked = true;

}

else

{

Female.Checked = true;

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

Student s = students[current];

s.id = textBox1.Text;

s.age = textBox2.Text;

s.marks = textBox3.Text;

if (Male.Checked)

{

s.gender = "Male";

}

else {

s.gender = "Female";

}

}

private void button3\_Click(object sender, EventArgs e)

{

students.RemoveAt(current);

current--;

}

private void oPENToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

private void eXITToolStripMenuItem1\_Click(object sender, EventArgs e)

{

this.Close();

}

private void eXITToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

StreamWriter sw = new StreamWriter(saveFileDialog1.FileName,true);

foreach (Student s in students)

{

sw.WriteLine(s.id + " " + s.age + " " + s.marks+ " "+s.gender);

}

sw.Close();

MessageBox.Show("File Saved Successfully");

}

}

private void saveFileDialog1\_FileOk(object sender, CancelEventArgs e)

{

}

private void sAVEToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

StreamReader sr = new StreamReader(openFileDialog1.FileName,true);

string line = sr.ReadLine();

students = new List<Student>();

while (line != null)

{

string[] tokens = line.Split();

Student s = new Student();

s.id = tokens[0];

s.age = tokens[1];

s.marks = tokens[2];

s.gender = tokens[3];

students.Add(s);

line = sr.ReadLine();

}

current = 0; } } }}

STUDENT CLASS:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab6

{

class Student

{

public string id;

public string age;

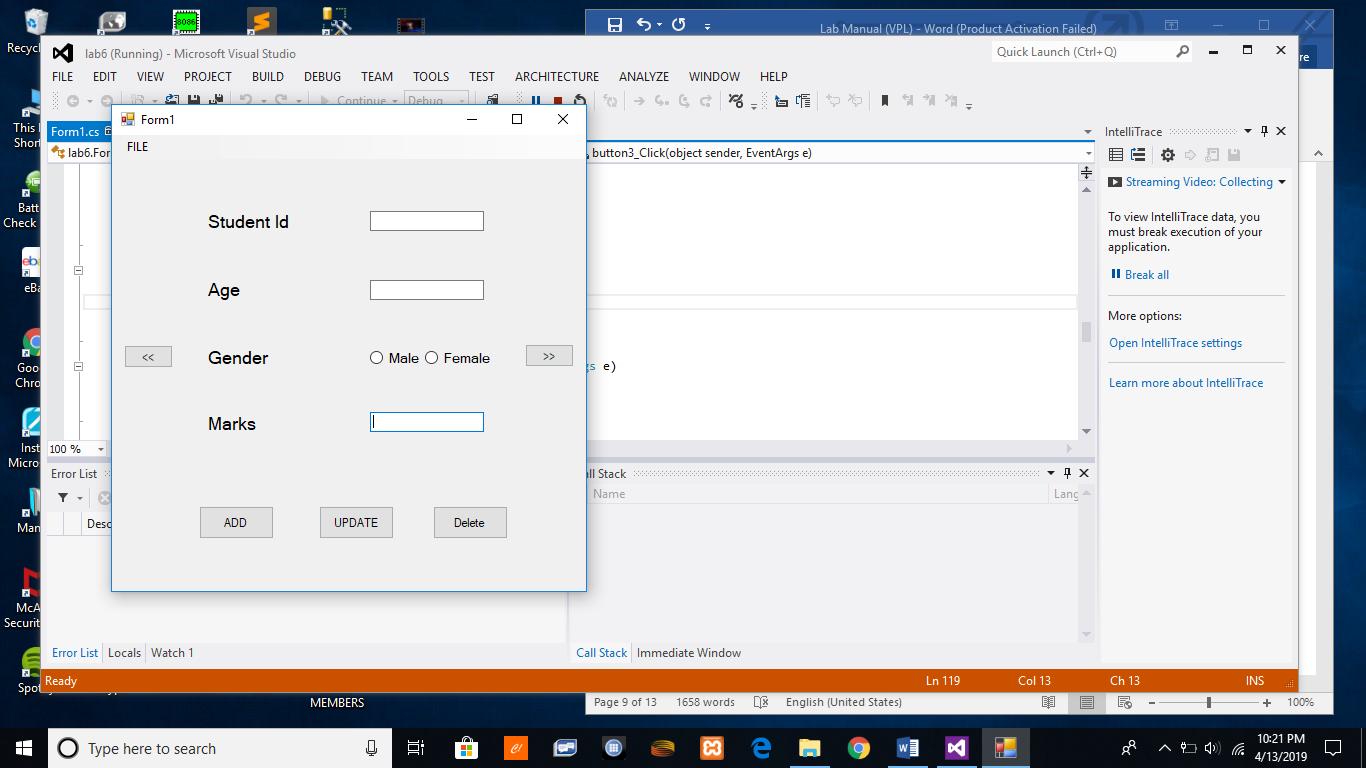
public string gender;

public string marks;

}

}

Output



**Lab 7: To study and implement XML parsing in C#**

Extensible Markup Language (XML) defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is a software- and hardware-independent tool for storing and transporting data

**Lab Tasks:**

1. Create a form with a text field centered on window.
2. Create a menu strip to load XML configuration file.
3. Create an XML file to store the following information:

* Window size: The size of the current window
* Window title: The title of the window
* Background color: The background color of the window
* Foreground color: The foreground color of the window

1. You should be able to apply the settings stored in XML file to the window.

Code

<?xml version="1.0"?>

<configuration>

<background>red</background>

<foreground>blue</foreground>

<text>hello</text>

<title>my title</title>

<width>100</width>

<height>100</height>

</configuration>

Code:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Xml;

namespace WindowsFormsApplication8

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void openToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

XmlDocument d = new XmlDocument();

d.Load(openFileDialog1.FileName);

XmlNodeList l = d.GetElementsByTagName("title");

this.Text = l[0].InnerText;

l = d.GetElementsByTagName("text");

this.textBox1.Text = l[0].InnerText;

l = d.GetElementsByTagName("width");

this.Width = int.Parse(l[0].InnerText);

l = d.GetElementsByTagName("height");

this.Height= int.Parse(l[0].InnerText);

l = d.GetElementsByTagName("background");

this.BackColor= Color.FromName(l[0].InnerText);

l = d.GetElementsByTagName("foreground");

this.textBox1.ForeColor = Color.FromName(l[0].InnerText);

this.label1.ForeColor = Color.FromName(l[0].InnerText);

}

}

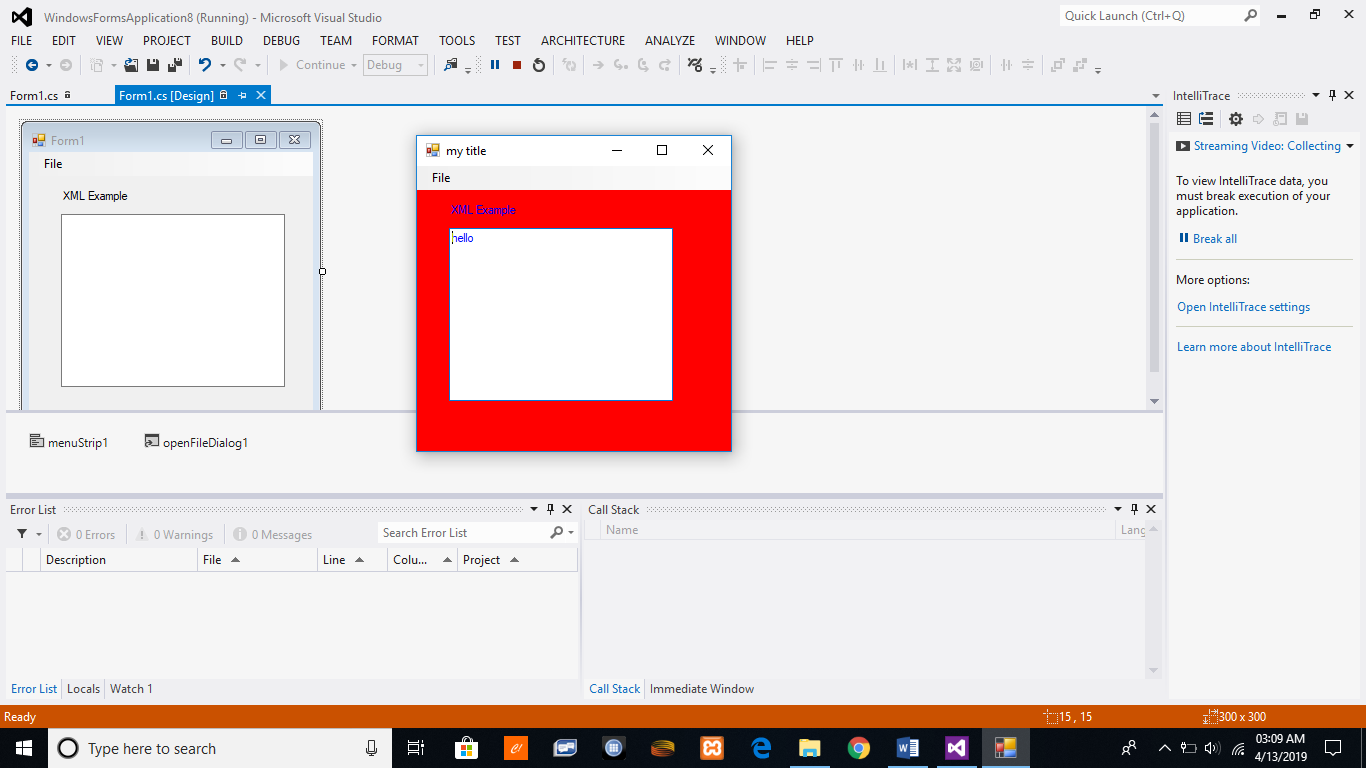
private void Form1\_Load(object sender, EventArgs e)

{

} }

}

Output



**Lab 8: To study and implement WPF and its layouts in C#**

In this lab, we will try to implement user interfaces using WPF and its layouts. We will also work with WPF data grid control.

**Lab Task:**

* 1. Design the following user interface in WPF using its various layouts.
  2. User should be able to add a new student as well as update and delete.
  3. The center data grid control should be resized when the window is maximized.

**Code**

<Window x:Class="WpfApplication1.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="MainWindow" Height="350" Width="525"

Loaded ="Window\_Loaded\_1" >

<Grid>

<DockPanel>

<Grid DockPanel.Dock="Top" HorizontalAlignment="Left">

<Button Content="update" Margin="355,57,47,43" ></Button>

<Button Content="cancel" Margin="355,90,47,10"></Button>

<Label Content="id" HorizontalAlignment="Left" Margin="103,25,0,0" VerticalAlignment="Top"/>

<Label Content="age" HorizontalAlignment="Left" Margin="103,77,0,0" VerticalAlignment="Top"/>

<Label Content="name" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="103,51,0,0"/>

<TextBox HorizontalAlignment="Left" Height="23" Margin="166,51,0,0" TextWrapping="Wrap" Text="TextBox" VerticalAlignment="Top" Width="120"/>

<TextBox HorizontalAlignment="Left" Height="23" Margin="166,25,0,0" TextWrapping="Wrap" Text="TextBox" VerticalAlignment="Top" Width="120"/>

<TextBox HorizontalAlignment="Left" Margin="166,79,0,17" TextWrapping="Wrap" Text="TextBox" Width="120"/>

</Grid>

<StackPanel HorizontalAlignment="Right" DockPanel.Dock="Bottom" Orientation="Horizontal">

<Button Margin="10,0,0,0" Content="Refresh"> </Button>

<Button Margin="10,0,0,0" Content="Delete"></Button>

</StackPanel>

<DataGrid x:Name="datagrid1" HorizontalAlignment="Stretch" Margin="0,0,0,0" VerticalAlignment="Stretch" SelectionChanged="DataGrid\_SelectionChanged\_1" />

</DockPanel>

</Grid>

</Window>

**Student class**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace WpfApplication1

{

class Sudent

{

public string id {set; get;}

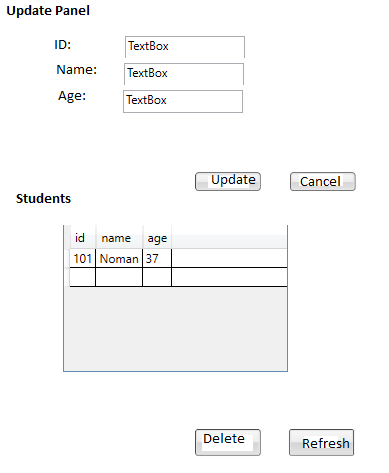
public string name {set; get;}

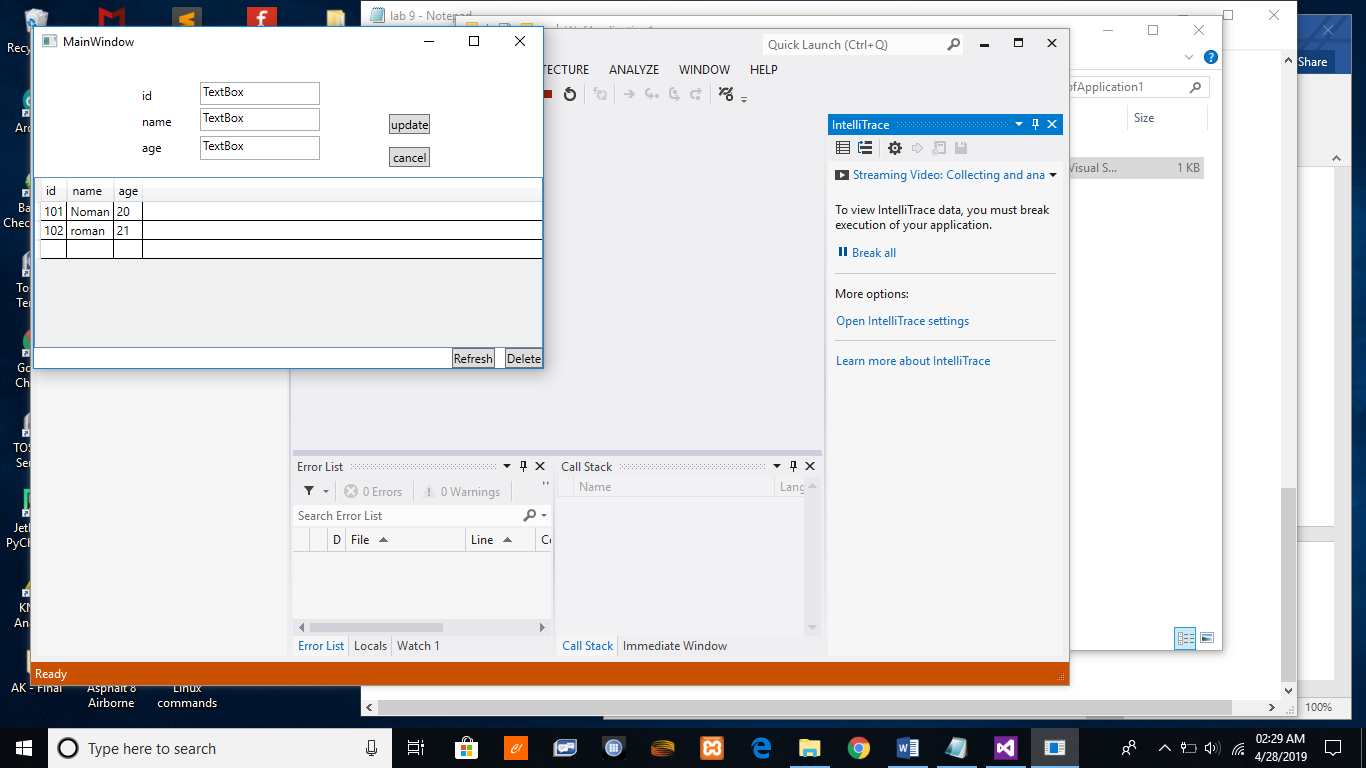
public string age {set; get;}

}

}

Output





**Lab 9: To study and implement LINQ in C#**

LINQ is an extension to the C# language that integrates data query directly into the programming language itself. Visual Studio 2015 and the .NET Framework 4.5 come with a number of built-in LINQ providers that provide query solutions for different types of data

* LINQ to Objects
* LINQ to XML
* LINQ to Entities
* LINQ to Data Set

**Lab Tasks:**

1. Create an array of 1000 randomly generated numbers. Use the LINQ query to find all the odd numbers from the list. Find the count of total odd numbers. Find the maximum and minimum odd number.
2. Create a class of Student with name, subject, and marks. Now add the students in a List. Using LINQ methods and group by, find the average marks of students.
3. Create a WPF project in which you can add doctor’s details such as name, qualification and salary. The added information is saved in a List. Provide a text area through which user can write LINQ query that can be run against the list.

**Code**

task 1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication16

{

class Program

{

static void Main(string[] args)

{

int[] nums = GenerateLotsOfNumbers(100);

Console.WriteLine("Numbers");

foreach (var i in nums)

{

Console.Write(i + "");

}

Console.WriteLine("\n odd");

var result = from n in nums

where n % 2 != 0

select n;

int count = result.Count();

int max = result.Max();

int min = result.Min();

Console.WriteLine("Count {0}, Max {1}, Min {2}",count,max,min);

foreach(var i in result)

{

Console.Write(i +"");

}

Console.ReadKey();

}

private static int[] GenerateLotsOfNumbers(int count)

{

Random generator = new Random(0);

int[] result = new int[count];

for (int i = 0; i < count; i++)

{

result[i] = generator.Next();

}

return result;

}

}

}

Task 2

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication17

{

class Program

{

static void Main(string[] args)

{

List<Student> stds = new List<Student>();

stds.Add(new Student ("A", "English",57));

stds.Add(new Student("B", "English", 58));

stds.Add(new Student("C", "English", 59));

var results = stds.GroupBy(s => s.name, (key, g) => new { Student = key, Average= g.Average(s=>s.marks)});

foreach(var v in results)

{

Console.WriteLine(v);

}

Console.ReadKey();

}

}

}

student class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication17

{

class Student

{

public string name { set; get; }

public string subject { set; get; }

public int marks { set; get; }

public Student(string name, string subject, int marks)

{

this.name = name;

this.subject = subject;

this.marks = marks;

}

}

}

**Lab 10: To study and implement ADO .Net in C#**

The .NET platform defines a number of namespaces that allow you to interact with relational database systems. Collectively speaking, these namespaces are known as ADO.NET. In this lab, we will use ADO .net to connect to database.

**Lab Tasks:**

1. Create a database named School. Create a table Student.
2. Now create a windows form and show all the students records in a GridView
3. In the same windows form allow the user to add, update and delete a student record.

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace WpfApplication2

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

List<Student> stds = new List<Student>();

public MainWindow()

{

InitializeComponent();

}

private void Window\_Loaded\_1(object sender, RoutedEventArgs e)

{

Student s;

stds.Add(s = new Student());

s.id = "101";

s.name = "Noman";

s.age = 20;

stds.Add(s = new Student());

s.id = "102";

s.name = "Saleem";

s.age = 21;

stds.Add(s = new Student());

s.id = "103";

s.name = "Najma";

s.age = 22;

datagrid1.ItemsSource = stds;

}

}

}

Code

Output

